CLAIMS:

What is claimed is:

1. A data processing system, comprising:
a plurality of hardware devices;
a plurality of operating systems; and
a firmware component for virtualizing the plurality of hardware devices for interaction with the plurality of operating systems; wherein
the firmware component is implemented using 64-bits.

2. The data processing system as recited in claim 1, wherein the plurality of hardware devices comprise a plurality of processors and wherein each of the plurality of processors operates in a 64-bit mode.

3. The data processing system as recited in claim 1, wherein the firmware component comprises a firmware kernel and the firmware kernel maintains a list of address and size pairs that describe cacheable system memory addresses.

4. The data processing system as recited in claim 1, wherein a primitive method checks addresses to determine whether the address is cacheable or cache-inhibited.

5. The data processing system as recited in claim 4, wherein the primitive method, responsive to a determination that the address is cacheable, carries out

the method using an appropriate machine language instruction.

6. The data processing system as recited in claim 4, wherein the plurality of hardware devices comprises a plurality of processors and wherein the primitive method, responsive to a determination that the address is cache-inhibited, enables a real mode cache-inhibited mechanism on one of the plurality of processors, allows access to the address to be performed by machine language instructions within the one of the plurality of processors, and disables the cache-inhibited mechanism.

7. The data processing system as recited in claim 1, wherein 32-bit values are zero-extended into 64-bit values.

8. The data processing system as recited in claim 1, wherein the firmware supports both 32-bit code and 64-bit code.

9. A method of providing a virtual copy of hardware resources within a data processing system to an operating system, the method comprising:

receiving a request to perform an action;

responsive to a determination that values associated with the request are 64-bit quantities, performing the request; and

responsive to a determination that the values associated with the request are 32-bit values, zero extending the values to 64-bit quantities and performing

the request using the 64-bit quantities.

10. The method as recited in claim 9, wherein the requested action is an arithmetic operation.

11. The method as recited in claim 9, wherein the requested action is an arithmetic comparison.

12. The method as recited in claim 9, wherein the requested action is a logical operation.

13. The method as recited in claim 9, further comprising:

responsive to a determination that the requested action is a cache-inhibited action, enabling a cache-inhibited mode within a processor, performing the requested action, and disabling the cache-inhibited mode.

14. The method as recited in claim 13, wherein a list of address and size pairs that describe cacheable system memory addresses are maintained and an address not falling within one of the address ranges within the list is considered to be a cache-inhibited address.

15. A computer program product in a computer readable media for use in a data processing system for providing a virtual copy of hardware resources within a data processing system to an operating system, the computer program product comprising:

first instructions for receiving a request to perform an action;

second instructions, responsive to a determination that values associated with the request are 64-bit quantities, for performing the request; and

third instructions, responsive to a determination that the values associated with the request are 32-bit values, for zero extending the values to 64-bit quantities and performing the request using the 64-bit quantities.

16. The computer program product as recited in claim 15, wherein the requested action is an arithmetic operation.

17. The computer program product as recited in claim 15, wherein the requested action is an arithmetic comparison.

18. The computer program product as recited in claim 15, wherein the requested action is a logical operation.

19. The computer program product as recited in claim 15, further comprising:

fourth instructions, responsive to a determination that the requested action is a cache-inhibited action, for enabling a cache-inhibited mode within a processor, performing the requested action, and disabling the cache-inhibited mode.

20. The computer program product as recited in claim 19, wherein a list of address and size pairs that describe cacheable system memory addresses are maintained and an address not falling within one of the address ranges within the list is considered to be a cache-inhibited

address.

21. A system for providing a virtual copy of hardware resources within a data processing system to an operating system, the system comprising:

first means for receiving a request to perform an action;

second means, responsive to a determination that values associated with the request are 64-bit quantities, for performing the request; and

third means, responsive to a determination that the values associated with the request are 32-bit values, for zero extending the values to 64-bit quantities and performing the request using the 64-bit quantities.

22. The system as recited in claim 21, wherein the requested action is an arithmetic operation.

23. The system as recited in claim 21, wherein the requested action is an arithmetic comparison.

24. The system as recited in claim 21, wherein the requested action is a logical operation.

25. The system as recited in claim 21, further comprising:

fourth means, responsive to a determination that the requested action is a cache-inhibited action, for enabling a cache-inhibited mode within a processor, performing the requested action, and disabling the cache-inhibited mode.

26. The system as recited in claim 25, wherein a list of address and size pairs that describe cacheable system memory addresses are maintained and an address not falling within one of the address ranges within the list is considered to be a cache-inhibited address.